Attorney Docket No. 42390.P6532 Patent

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:

Maliszewski

Serial No. 09/217,498

Filed: December 21, 1998

For: METHOD AND APPARATUS TO TEST AN INSTRUCTION SEQUENCE

Examiner: W. Lin

Art Unit: 2154

PRELIMINARY AMENDMENT

ATTN: BOX PATENT APPLICATION
ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Sir:

In response to the Office Action mailed December 4, 2000, Applicant submits this amendment and response for consideration. Please consider the following:

IN THE CLAIMS

(Un-annotated versions of the amended claims follow these annotated versions)

1. (Second Amendment) A method comprising:

generating a software test module to produce a test result by performing a test on [the sequence of] instructions;

in the [sequence of] instructions, replacing a first instruction comprising a target address with a second non-identical instruction having an instruction address in the [sequence] instructions, the second instruction to transfer control to the test module; and

storing the target address encrypted in a table, the test module to locate the target address in the table and to set an execution address to the target address if the test result indicates the [sequence of] instructions [is] are to proceed.

2. (First Amendment) The method of claim 1 further comprising

compacting the [sequence of] instructions to eliminate a hole created by replacing the first instruction with the second instruction.

4. The method of claim 1 further comprising:

profiling the [sequence of] instructions to identify the first instruction as an instruction to replace.

5. (Second Amendment) A device comprising:

a processor;

a machine-readable storage medium coupled to the processor by way of a bus, the storage medium having stored thereon [a sequence of] instructions which, when executed by the processor, cause the data processing device to

[generating] generate a software test module, the [generated] test module to produce a test result by performing a test on the [sequence of] instructions;

in the [sequence of] instructions, replace a first instruction comprising a target address with a second non-identical instruction having an instruction address in the [sequence] instructions, the second instruction to transfer control to the test module; and

store the target address in an encrypted table, the test module to locate the target address in the table and to transfer control to the target address if the test result indicates the [sequence of] instructions [is] are to proceed.

6. (First Amendment) The device of claim 5 in which the [sequence of] instructions, when executed by the processor, further cause the device to:

compact the [sequence of] instructions to eliminate a hole created by replacing the first instruction with the second instruction.

7. (First Amendment) The device of claim 5 in which the [sequence of] instructions, when executed by the processor, further cause the device to:

corresponding the target address with the instruction address in the encrypted table.

8. (First Amendment) The device of claim 5 in which the [sequence of] instructions, when executed by the processor, further cause the device to:

profile the [sequence of] instructions to identify the first instruction as an instruction to replace.

9. (First Amendment) An article comprising:

a machine-readable medium having stored thereon [a sequence of] instructions which, when executed by a data processing device, cause the data processing device to:

generating a software test module to produce a test result by performing a test on the [sequence of] instructions;

in the [sequence of] instructions, replace a first instruction comprising a target address with a second non-identical instruction having an instruction address in the [sequence] instructions, the second instruction to transfer control to the test module; and

store the target address in an encrypted table, the test module to locate the target address in the table and to transfer control to the target address if the test result indicates the [sequence of] instructions [is] are to proceed.

10. (First Amendment) The article of claim 9 in which the [sequence of] instructions, when executed by a data processing device, further cause the data processing device to:

compact the [sequence of] instructions to eliminate a hole created by replacing the first instruction with the second instruction.

11. (First Amendment) The article of claim 9 in which the [sequence of] instructions, when executed by a data processing device, further cause the data processing device to:

correspond the target address with the instruction address in the encrypted table.

12. (First Amendment) The article of claim 9 in which the [sequence of] instructions, when executed by a data processing device, further cause the data processing device to:

profile the [sequence of] instructions to identify the first instruction as an instruction to replace.

13. (First Amendment) An article comprising:

a machine-readable medium having stored thereon:

[a sequence of] instructions which, when executed by a data processing device, cause the data processing device to:

transfer control to a software test module when a second instruction having an instruction address in the [sequence] instructions is executed by the data processing device, the second instruction replacing a non-identical first instruction comprising a target address;

a test module, the test module comprising

a table comprising a target address of the replaced first instruction; and

test instructions to produce a test result by performing a test on the [sequence of] instructions, the test module to locate the target address in the table and to transfer control to the target address if the test result indicates the [sequence of] instructions [is] are to proceed.

14. (First Amendment) The article of claim 13 in which the [sequence of] instructions further comprise[s]

instructions to load the test module.

16. (Second Amendment) The article of claim 14 in which the test module further comprises:

instructions moved from the [sequence of] instructions, the instructions moved to make room in the [sequence of] instructions for the instructions to load the test module.

(Un-annotated)

1. A method comprising:

generating a software test module to produce a test result by performing a test on instructions;

in the instructions, replacing a first instruction comprising a target address with a second non-identical instruction having an instruction address in the instructions, the second instruction to transfer control to the test module; and

storing the target address encrypted in a table, the test module to locate the target address in the table and to set an execution address to the target address if the test result indicates the instructions are to proceed.

2. The method of claim 1 further comprising

compacting the instructions to eliminate a hole created by replacing the first instruction with the second instruction.

4. The method of claim 1 further comprising:

profiling the instructions to identify the first instruction as an instruction to replace.

5. A device comprising:

a processor;

a machine-readable storage medium coupled to the processor by way of a bus, the storage medium having stored thereon instructions which, when executed by the processor, cause the data processing device to

generate a software test module, the test module to produce a test result by performing a test on the instructions;

in the instructions, replace a first instruction comprising a target address with a second non-identical instruction having an instruction address in the instructions, the second instruction to transfer control to the test module; and

store the target address in an encrypted table, the test module to locate the target address in the table and to transfer control to the target address if the test result indicates the instructions are to proceed.

6. The device of claim 5 in which the instructions, when executed by the processor, further cause the device to:

compact the instructions to eliminate a hole created by replacing the first instruction with the second instruction.

7. The device of claim 5 in which the instructions, when executed by the processor, further cause the device to:

corresponding the target address with the instruction address in the encrypted table.

8. The device of claim 5 in which the instructions, when executed by the processor, further cause the device to:

profile the instructions to identify the first instruction as an instruction to replace.

9. An article comprising:

a machine-readable medium having stored thereon instructions which, when executed by a data processing device, cause the data processing device to:

generating a software test module to produce a test result by performing a test on the instructions;

in the instructions, replace a first instruction comprising a target address with a second non-identical instruction having an instruction address in the instructions, the second instruction to transfer control to the test module; and

store the target address in an encrypted table, the test module to locate the target address in the table and to transfer control to the target address if the test result indicates the instructions are to proceed.

10. The article of claim 9 in which the instructions, when executed by a data processing device, further cause the data processing device to:

compact the instructions to eliminate a hole created by replacing the first instruction with the second instruction.

11. The article of claim 9 in which the instructions, when executed by a data processing device, further cause the data processing device to:

correspond the target address with the instruction address in the encrypted table.

12. The article of claim 9 in which the instructions, when executed by a data processing device, further cause the data processing device to:

profile the instructions to identify the first instruction as an instruction to replace.

13. An article comprising:

a machine-readable medium having stored thereon:

instructions which, when executed by a data processing device, cause the data processing device to:

transfer control to a software test module when a second instruction having an instruction address in the instructions is executed by the data processing device, the second instruction replacing a non-identical first instruction comprising a target address;

a test module, the test module comprising

a table comprising a target address of the replaced first instruction; and

test instructions to produce a test result by performing a test on the instructions, the test module to locate the target address in the table and to transfer control to the target address if the test result indicates the instructions are to proceed.

14. The article of claim 13 in which the instructions further comprise

instructions to load the test module.

16. The article of claim 14 in which the test module further comprises:

instructions moved from the instructions, the instructions moved to make room in the instructions for the instructions to load the test module.

		Office must recognized that such a strained interpretation of the act of "replacing" will not stand on appeal. It is improper and illogical to assert, as the Office does, that the plain meaning of the claim terms "a first instruction" and "a second instruction" refer to the same identical instruction. However, Applicant has amended the claims so that no further disagreement on this point can occur.	
store the target address in an encrypted table, the test module to locate the target address in the table and to transfer control to the target address if the test result indicates that the instructions are to proceed.	The DSP call instruction may include a displacement field which comprises an index into a table of DSP routine target addresses [Col. 11 lines 30-40].	The claims recite that control is transferred to the target address if the test result indicates that instructions are to proceed. No such teaching may be found in any of the cited references. Mills teaches that control is always transferred to either the target address of the DSP routine or to the emulation routine. The "test result" of Mills comprises an indication of whether or not the DSP is present and enabled. If present, control is transferred to the DSP routine; otherwise, to the emulation routine. No determination is made by the decoder of Mills as to whether or not the instructions are to proceed. Such teaching is simply absent.	
Claim 14	Corresponding Element in Prior Art Cited By Office Action	Analysis	Notes
The article of claim 13 in which the instructions further comprise instructions to load the test module into a	Mills taught that the sequence of instructions further comprises instructions to load the test module[i.e., the	The Office is now apparently identifying the instructions themselves as the test module. If the Office is	The Office must clarify what elements of the cited references are relied upon as comprising the "test

REMARKS

In an Office Action mailed December 4, 2000, claims 1-12 are rejected under 35 U.S.C. 103 as unpatentable over Mills et al (U.S. Patent 5,721,945 – henceforth Mills) in view of Bianco (U.S. Patent 5,386,471 – henceforth Bianco). Claims 13-16 are rejected under 35 U.S.C. 102(e) as anticipated by Mills. Please consider the following remarks in light of the amended claims. Note that references herein to "the Office" refer to the Examiner in his capacity as representative of the Patent and Trademark Office.

Mills teaches a system including a processor and a digital signal processor (DSP). The DSP and processor cooperate by way of a DSP call instruction. When the DSP is present in the system, the call instruction invokes a function of the DSP. Otherwise, the DSP call is emulated by the processor. The target address is either the starting address of a routine in the processor instruction set, or the starting address of a routine in the DSP instruction set [Col. 4, lines 6-46]. If the DSP is present the target address is used to invoke the DSP to execute the DSP routine. Otherwise the target address is used to invoke the processor routine via a processor subroutine call [Col. 5 lines 37-52]. The DSP call instruction may include a displacement field which comprises an index into a table of target addresses [Col. 11 lines 30-40].

There is a great deal of confusion and apparent contradiction is the Office Action concerning what specifically is identified as the test module in the cited prior art. To clarify what the Office Action is apparently treating as the "test module" in the prior art, Applicant provides the following table.

Claim 1	Corresponding Element in Prior Art Cited By the Office	Analysis	Notes
generating a test module to produce a test result by performing a test on instructions;	the decode unit performs a test on the sequence of instructions being decoded to see if there is a DSP call instruction in the sequence	The Office cites the decode unit as performing the test on the instructions, e.g. the decode unit is the test module. However this	The claims have been amended to indicate the test module is software. None of the references teach a software test module.

		assertion is contradicted below. Note that the claims recite more than merely performing a test, e.g. generating the test module. There is simply no teaching whatsoever in Mills of generating the test module (decode unit) which performs the test.	
in the instructions, replacing a first instruction comprising a target address with a second instruction having an instruction address in the instructions, the second instruction to transfer control to the test module;	As noted, the Office previously equates the decode unit to the test module, e.g. the thing that performs the test. Here, however, the Office asserts that the DSP instruction, when executed, could cause transfer of control to either the DSP core or a DSP subroutine, "wherein both are parts of the test module". Here, the Office relies upon both the DSP core and the DSP subroutine being part of the test module, e.g. the decoder, the thing that performs the test. There is simply no teaching, suggestion, or inference in Mills that the decode unit comprises the DSP core and the DSP subroutine. Mills teaches exactly the opposite, that the core and decoder are distinctly separate. The teaching of Mills is clear: the decode unit tests the sequence for a DSP call instruction, and the DSP core comprises DSP code which may be invoked by that call instruction.	It is a clear contradiction to first assert that the decode unit is the test module to meet one limitation of the claim, and then in the subsequent sentence to assert the opposite, e.g. that the DSP core is the test module. Mills simply does not teach the DSP core performing any test whatsoever on the instructions. It cannot be the test module, and it is not part of the decoder. Such teaching is simply absent from the prior art and cannot be relied upon as a basis of rejection. Mills does not teach instruction replacement. Mills teaches that the decoder transfers control to either a DSP routine or an emulation routine, based upon whether or not the DSP core is present and enabled. No act of instruction replacement occurs. How can the Office assert that the act of "replacement" is met by leaving the sequence completely unaltered (or, as the Office describes it, "replacing the [instruction] by itself"? Surely the	The claims have been amended to indicate that the first and second instructions are non-identical.

memory.	instructions to be decoded by the decoding unit and to be executed by the DSP core are the instruction to load the test module.	referring to the decoder as the test module, the references fail to demonstrate how instructions can load a decoder. Rather, it is well established in the art that the opposite is true - a decoder loads instructions (to decode them).	module".

Regarding claims 2, 6, and 10, compaction to eliminate holes produced by replacing a first instruction with a second instruction is not well known in the art such that Official Notice may be taken. Producing a reference which provides such teaching would be trivial if indeed such practice were so well known, and yet the Office has provided none.

Regarding claims 4, 8, and 11, profiling the instructions to identify a first instruction to replace with a second instruction is not well known in the art such that Official Notice may be taken. Producing a reference which provides such teaching would be trivial if indeed such practice were so well known, and yet the Office has provided none.

Regarding claim 15, the Office asserts that the decode unit treats the DSP call instruction as a typical subroutine call and transfers control to the execution unit for DSP simulation. The Office asserts that this is an equivalent mechanism to an exception handler. The Office is incorrect as a matter of technology. An exception handler invokes the exception handling capabilities of the execution platform, which, for example, may involve software or hardware interrupts, among other techniques. Exception handling involves setting a handler and generating an exception; a mere subroutine call does not.

In light of these argument and the amendments, Applicant believes that all claims are in condition for allowance. Applicants respectfully request allowance of all claims.

Dated: 1-16-01

Charles A. Mirho
Registration No. 41,199

c/o Blakely, Sokoloff, Taylor & Zafman, LLP
12400 Wilshire Blvd.
Seventh Floor
Los Angeles, CA 90025-1026
(503) 264-0967

"Express Mail" mailing label number EL034439825US
Date of Deposit JANUARY 17, 2001
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231

Signature 1/17/01
Date